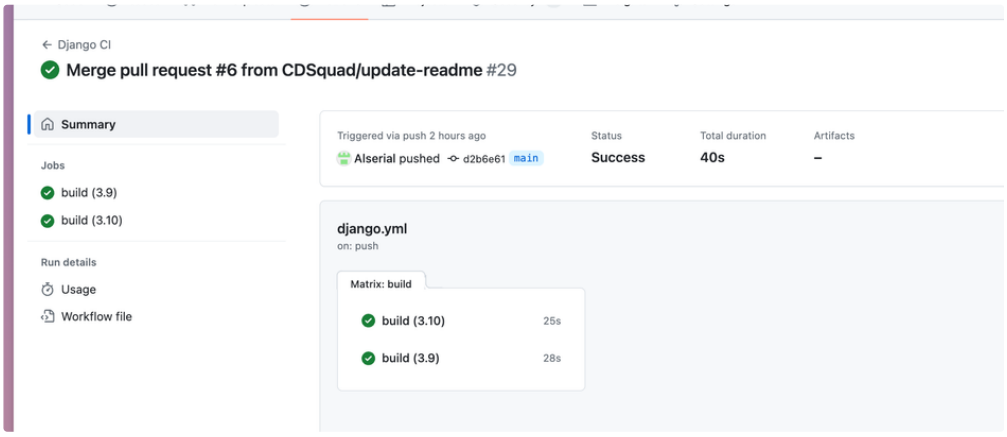
Deployment

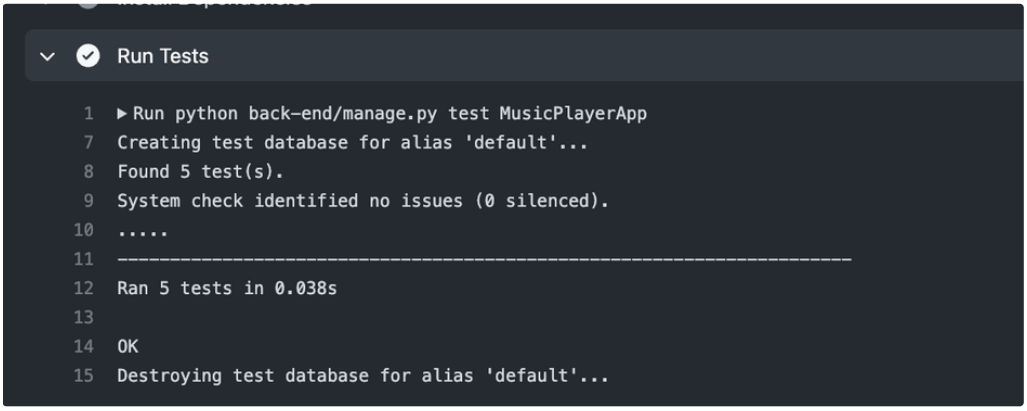
Documentation for MusicPlayerApp

1. **Introduction**: This document outlines the procedures for the continuous integration (CI) and deployment of the MusicPlayerApp, a Django-based application. The CI process is facilitated through GitHub Actions, and the application is composed of both backend and frontend components. The backend is manually deployed to PythonAnywhere due to the restrictions of the free plan, while the frontend is hosted on GitHub Pages.
2. **Continuous Integration (CI)**: 2.1 **Overview**: We utilize GitHub Actions for our CI pipeline. This system automatically triggers build and test processes upon pushes or pull requests to the "main" branch. The CI is configured to work with Python versions 3.9 and 3.10.

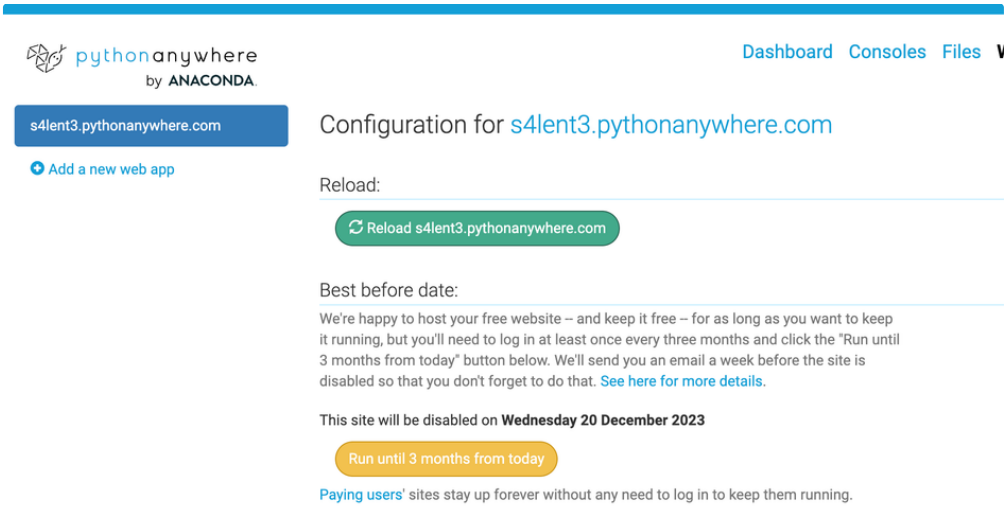
   2.2 **Workflow**: The following steps are executed in our CI pipeline:
   - Current directory printout
   - Repository checkout
   - Python setup
   - Dependency installation
   - Unit testing

   2.3 **Unit Testing**: Tests are written using Django's testing framework and cover various aspects of the application, including user management, music management, and file uploads. The tests ensure that all components of the application work as expected and that new changes do not introduce regressions.

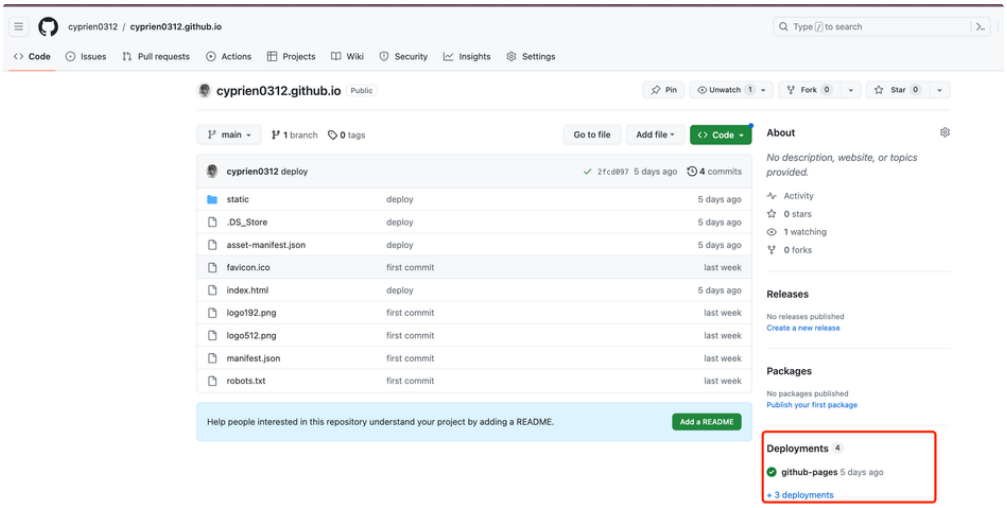




3. **Deployment**: 3.1 **Backend**: The backend is hosted on PythonAnywhere. Due to the limitations of the free plan, continuous deployment (CD) is not feasible because it lacks SSH access. As such, updates to the backend are carried out manually. This approach is currently deemed sufficient as backend updates are infrequent.



3.2 **Frontend**: The frontend is statically hosted via GitHub Pages at `https://github.com/cyprien0312/cyprien0312.github.io`. The process is straightforward, as GitHub Pages directly serves the files pushed to the repository. The deployment updates automatically upon new commits to the repository.



4. **Procedures**: 4.1 **Manual Backend Update**: When the backend needs an update, the following manual procedure is followed: a. A team member pulls the latest changes from the "main" branch. b. They log in to the PythonAnywhere console. c. They navigate to the application's directory. d. They upload the new files or changes via the PythonAnywhere file upload interface or update the code directly via the built-in editor. e. They reload the application to apply the changes.

   4.2 **Frontend Update**: Frontend updates are automated. Any push or merge to the main branch of the `cyprien0312/cyprien0312.github.io` repository triggers a deployment. GitHub Pages automatically builds and deploys the new changes, making them live almost immediately.

5. **Future Considerations**:
   - Explore budget-friendly options for a paid PythonAnywhere plan to enable continuous deployment through SSH.
   - Consider additional monitoring solutions to ensure the application's health and performance.